Course Staff and Schedule

Instructor  Joe Zachary
Office: MEB 3190a
Phone: 801-581-7079
Email: zachary@cs.utah.edu
Office Hours: Tue/Thu 1:00-5:00

Lectures  MW 3:00-4:20, ASB 220

TAs  Stephen DeBies  Dallin Van Mondfrans
     Ryan Cantera  Peter Pan
     Safia Hassan  Pavithra Pappa
     Ashlee Kurys  Scott Steadham
     Jac MacHardy  Andrew Vuong

Lab 7  T 8:35-9:25  On the first two (possibly three) Tuesdays
2  T 9:40–10:30  of the semester, the labs will meet in
3  T 10:45–11:35  WEB L130. The rest of the semester
4  T 11:50–12:40  the labs will meet in MEB 3225.
5  T 12:55–1:45
6  T 2:00–2:50
8  T 3:05-3:55
9  T 4:10-5:00
10  T 5:15-6:05
11  T 6:20-7:10

Course Overview

You will learn how to write computer programs using Java. We will highlight specification, design, implementation, and testing. By the end of this course, you will be able to start with a problem statement and then design, implement, document, and test an object-oriented program that solves the problem.

You will learn to think like a computer scientist. The goal is to give you the knowledge, intuition, and habits on which you can base a lifetime of learning. We will discuss decomposition and abstraction as
powerful habits of thought; the roles of procedures, abstract data types, and polymorphism in organizing programs; and the importance of specifications and unit testing in program design and implementation.

You will learn about key ideas in computer science. We will discuss command-line and graphical user interfaces; text processing and graphics; iteration and recursion; lists, sets, and maps.

The prerequisite for this course is CS 1030, which can be waived (via an assessment test) for students with sufficient experience. Students taking CS 1410 should have completed (or currently be taking) a calculus course.

Pragmatics

Lectures. We will meet for lecture on Mondays and Wednesdays, 3:00-4:20. In lecture, I will make use of a projected computer. After lecture, I will post to Canvas any material that I projected (such as Java programs or slides). I will also post a video capture of my computer screen as well as a video of the lecture.

Textbook. Instead of a traditional textbook, we will be using a “zyBook” titled Introduction to Object-Oriented Programming. A zyBook is an online textbook augmented with lots of interactive activities. The cost to access it for the semester is $48; directions for subscribing are linked from Canvas.

Computers. You will be spending a lot of time writing programs. When working on assignments, you can use your own computer or you can use one of the computers in the College of Engineering’s CADE Lab. Information on installing the required free software to your Windows, Macintosh, or Linux computer is available from Canvas, as is information on accessing the computers in the CADE Lab.

Labs. You will meet with two TAs in small laboratory sections for 50 minutes on Tuesdays. For the first two (or possibly three) weeks of the semester the labs will meet in WEB L130. For the remainder of the semester the labs will meet in WEB 3225. You will need to bring your own laptop to these labs. If you do not have a laptop, there will be a few loaners available to use during the lab sessions.

In your section, a TA will guide you through a set of interactive exercises related to programming or programming tools, review for exams, discuss the results of exams, or answer questions as you work on assignments. The laboratory sections will be absolutely essential to your success in the course, and you should plan on attending them all.

zyBook Participation Activities. In advance of each lecture, I will post to Canvas the zyBook sections that are assigned for that lecture. Prior to the beginning of the lecture, you must read the assigned sections and complete all of the participation activities that they contain.

zyBook Challenge Activities. By 11:59 p.m. each Friday, you must complete all of the challenge activities contained in the zyBook sections that were assigned for that week’s lectures.

Problem Sets. Most Fridays I will post to Canvas a problem set (a programming assignment) that will be due following Thursday. You will submit your solution via Canvas. Our goal is to have each problem set graded by the Wednesday following the due date. If you believe that a mistake was made in grading your assignment, you must let the grader know about the problem within one week. Use Canvas to message the TA who graded your assignment. Explain the issue clearly and carefully, and be prepared to respond to any questions that the grader might have.

Problem Set Late Policy. The weekly problem sets will be due at 11:59 pm on Thursdays. During the course of the semester, you may submit up to four problem sets up to 24 hours late without penalty. You do not need to obtain prior permission to use these four late days. No other late assignments will be accepted unless you obtain prior permission or some late-breaking emergency occurs.

Exams. The midterm will be on October 18, in place of lecture. The final will be on Tuesday, December 12 from 3:30–5:30 p.m. in the regular lecture room.
Schedule. Canvas contains a day-to-day schedule that will show upcoming lecture topics, zyBook assignments, labs, and problem sets. Following each lecture, I will update the schedule to reflect what was actually covered in lecture that day, and I will post the lecture material I used and the lecture videos I recorded.

Consulting Hours. The course staff (instructor and teaching assistants) will hold regular consulting hours each week, during which we will be available to help you with questions or problems. The consulting schedule has not yet been determined, but will be posted on Canvas soon.

Java and Eclipse. All of the programming that you do in this course will be in Java, using the Eclipse programming environment. Java and Eclipse are already installed on all CADE Lab computers. You can obtain free copies of Java and Eclipse to install on your own computer. Download links and instructions are available on Canvas.

Grading. Your final course grade will be based on your lab attendance, your zyBook participation and challenge activities, your problem sets, and your exams.

- Your attendance in the weekly lab sessions will be recorded. You can have automatic excused absences from up to three of those sessions. Your lab average will be the percentage of lab sessions that you attended or were excused.

- You will receive participation points for correctly completing participation activities that appear in the zyBook sections that are assigned for a lecture. Specifically, you will receive one participation point for each part of each such activity that you correctly complete before the lecture begins. Your participation average will be the percentage of the maximum possible score that you attain, multiplied by 1.2, with a max of 100.

- You will receive challenge points for correctly completing challenge activities that appear in the zyBook sections that are assigned for lectures. Specifically, you will receive one challenge point for each part of each such activity that you correctly complete by 11:59 p.m. on the Friday of the week of the lecture. Your challenge average will be the percentage of the maximum possible score that you attain, multiplied by 1.2, with a max of 100.

- Your problem set average will be the weighted average of your individual problem set grades. (Some problem sets may be worth more than others.)

- Your exam average will be the weighted average of your midterm (40%) and final (60%) scores.

Your exam and problem set averages must both be at least 55% or your course grade will be no higher than a D-, regardless of how high your overall average is.

Otherwise, grading will be done by calculating a weighted average of your programming assignment average (42%), your zyBook participation and challenge averages (6% each), your lab average (4%), and your exam average (42%). Course grades will be assigned on a curve consistent with the weighted averages of all the students.

Something Due Each Day

You have probably noticed that you will have something due each day of most weeks. On most Mondays and Wednesdays, you will have zyBook participation activities due. On Tuesdays you will a laboratory to attend. On most Thursdays you will have a substantial programming assignment coming due. On Fridays you will have zyBook challenge activities due.

Learning to program requires lots of thought and practice, and many students don’t give it the attention it requires. This due date schedule is designed to address this by keeping you actively thinking about the course. Don’t worry though; the schedule is not as onerous as it might first appear.
Each zyBook assignment will introduce topics that will be discussed in an upcoming lecture. The system gives you helpful feedback if you answer an activity incorrectly, and you can answer each question as many times as you like until you get it right.

Each lab activity is designed to be completed during your 50-minute lab session. In any event, you are graded only on attendance.

The weekly problem sets will require a lot of your time. A sure recipe for failure is to start working on a problem set the day it is due. It is much better to work on your problem set a little each day.

**How To Approach This Class**

For most of you, this will be your first or second class in computer science. Here is some advice on how to approach this class.

- Do the assigned zyBook participation activities before you come to lecture. The lectures do not duplicate the reading; they build on it. If you delay doing the reading and participation activities, you won’t understand the context of the lecture. You’ll also hurt your grade.

- Concentrate in lecture. The concepts that are presented in lecture are what’s important; you will be able to find the details in the zyBook. If you zone out with the idea of watching the video later, you are wasting your time coming to lecture. Instead, think about what is being said. Try to answer all the questions that are asked, even if only in your head. Raise your hand and ask a question when you don’t understand something. Try to understand everything. Don’t give up!

- Participate in the labs. You’ll be sitting at a laptop in a room with a teaching assistant and a few other students. Take advantage of the laptop by trying things out. That way, you’ll discover the things that you don’t understand in a setting where there are plenty of other people (TAs as well as students) to help you out.

- Respect the programming assignments. Some students expect that if they have done the zyBook activities, concentrated in lecture, and participated in the labs, then the assignments will be straightforward. What these students don’t understand is that the assignments are designed to challenge you by requiring that you apply the concepts you have learned to new situations. The assignments will be your most important learning experience in the course; they will rarely be straightforward. You should start each assignment as soon as it comes out. This way, you will have time to take a break when you get stuck.

- Don’t rely too much on the help of TAs or other students. Every semester there are students who get through the assignments by constantly asking the TAs for help. Such students ask for help immediately every time they get stuck, instead of working things out for themselves. As a result, these students don’t learn very much and typically fail the exams and thus the course.

Beginners are often surprised by the amount of human effort that has to go into designing, writing, and testing a program. Complaints from students about the amount of time required by introductory computer programming courses are universal. You should expect to spend three hours outside of class for every hour that you spend in class. In other words, you should expect to spend 12 hours per week reading, studying, doing zyBook activities, and designing and writing programs. Some of you will spend less, but many of you will spend even more time. Please keep this in mind when setting up your schedule for the semester!
Getting Help and Information

Canvas will contain a variety of resources, including course staff consulting hours and contact information, announcements, discussion forums, assignments, quizzes, examples from lecture, grades, and links pertaining to the reading, Java, and Eclipse.

All of the course staff will be available outside of formal classes to answer your questions and help with problems. We will post the consulting schedule to Canvas as soon as it is finalized. We encourage you to seek us out whenever you need help, advice, or encouragement.

When we wish to communicate with the entire class, we will post announcements to Canvas. Be sure to set your Canvas notification preferences so that you receive announcements via email. The emails that Canvas generates go to your UMail address by default. Be sure that you read your UMail regularly, forward your UMail to an account that you do read regularly, or change your Canvas email settings.

There is also a discussion forum in Canvas. Please use the discussion forum to ask questions that are likely to be of general interest to other students. For example, if you don’t understand an assignment, have a question about some aspect of Java, or want help with using Eclipse, post to the forum. That way, other students will be able to benefit from the answer. You should, of course, search through old postings before posting a new question.

When you wish to ask the entire course staff a question, send email to help-cs1410@lists.utah.edu. Do this (instead of posting to the discussion forum) only for questions that are either confidential or are unlikely to be of general interest to other students. For example, if you need to ask about your particular solution to an assignment, contact us this way.

If you need to contact one of us in particular, simply message us through Canvas.

Cooperation vs. Cheating

Working with others on assignments is a good way to learn the material and we encourage it. However, there are limits to the degree of cooperation that we will permit.

When working on programming assignments, you must work only with others whose understanding of the material is approximately equal to yours. In this situation, working together to find a good approach for solving a programming problem is cooperation; listening while someone dictates a solution is cheating. You must limit collaboration to a high-level discussion of solution strategies, and stop short of actually writing down a group answer. Anything that you hand in, whether it is a written problem or a computer program, must be written in your own words. If you base your solution on any other written solution, you are cheating.

It is OK to help other students, within limits. If you are asked for help by another student, two things that are absolutely forbidden are to show that student your solution or to put your hands on that student’s keyboard. That isn’t helping; that is facilitating cheating! Instead, answer questions; give tips; help with tools; explain Java; point out a bug; give encouragement. In other words, interact with other students the way that the TAs do.

When taking an exam, you must work completely independently of everyone else. Any collaboration here, of course, is cheating.

We do not distinguish between cheaters who copy others’ work and cheaters who allow their work to be copied.

If you cheat, you will be given an E in the course and referred to the University Student Behavior Committee. If you have any questions about what constitutes cheating, please ask.
Students With Disabilities

The University of Utah seeks to provide equal access to its programs, services and activities for people with disabilities. If you will need accommodations in the class, reasonable prior notice needs to be given to the Center for Disability Services, 162 Olpin Union Building, 581-5020 (V/TDD). CDS will work with you and the instructor to make arrangements for accommodations.

All written information in this course can be made available in alternative format with prior notification to the Center for Disability Services.
# Tentative Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/21</td>
<td>Introduction</td>
<td>10/16</td>
<td>Midterm Exam Review</td>
</tr>
<tr>
<td>8/22</td>
<td>Lab 1: Eclipse</td>
<td>10/17</td>
<td>Lab 8: Sample midterm discussion</td>
</tr>
<tr>
<td>8/23</td>
<td>Expressions</td>
<td>10/18</td>
<td>Midterm Exam</td>
</tr>
<tr>
<td>8/24</td>
<td>No PS due</td>
<td>10/19</td>
<td>No PS due</td>
</tr>
<tr>
<td>8/28</td>
<td>Static methods</td>
<td>10/23</td>
<td>Recursion</td>
</tr>
<tr>
<td>8/29</td>
<td>Lab 2: Static methods</td>
<td>10/24</td>
<td>Lab 9: Midterm exams returned</td>
</tr>
<tr>
<td>8/30</td>
<td>Objects</td>
<td>10/25</td>
<td>Recursion</td>
</tr>
<tr>
<td>8/31</td>
<td>PS1 due</td>
<td>10/26</td>
<td>PS7 due</td>
</tr>
<tr>
<td>9/4</td>
<td><strong>Labor Day Holiday</strong></td>
<td>10/30</td>
<td>Graphical User Interfaces</td>
</tr>
<tr>
<td>9/5</td>
<td>Lab 3: Eclipse debugger</td>
<td>10/31</td>
<td>Lab 10: Building GUIs</td>
</tr>
<tr>
<td>9/6</td>
<td>Conditionals</td>
<td>11/1</td>
<td>Graphical User Interfaces</td>
</tr>
<tr>
<td>9/7</td>
<td>PS2 due</td>
<td>11/2</td>
<td>PS8 due</td>
</tr>
<tr>
<td>9/11</td>
<td>Loops</td>
<td>11/6</td>
<td>Java memory model</td>
</tr>
<tr>
<td>9/12</td>
<td>Lab 4: Graphics objects</td>
<td>11/7</td>
<td>Lab 10: Memory model</td>
</tr>
<tr>
<td>9/13</td>
<td>Loops</td>
<td>11/8</td>
<td>Java collection classes</td>
</tr>
<tr>
<td>9/14</td>
<td>PS3 due</td>
<td>11/9</td>
<td>PS9 due</td>
</tr>
<tr>
<td>9/18</td>
<td>Arrays and ArrayLists</td>
<td>11/13</td>
<td>Polymorphism</td>
</tr>
<tr>
<td>9/19</td>
<td>Lab 5: Testing methods with JUnit</td>
<td>11/14</td>
<td>Lab 11: Polymorphism</td>
</tr>
<tr>
<td>9/20</td>
<td>Arrays and ArrayLists</td>
<td>11/15</td>
<td>Polymorphism</td>
</tr>
<tr>
<td>9/21</td>
<td>PS4 due</td>
<td>11/16</td>
<td>PS10 due</td>
</tr>
<tr>
<td>9/25</td>
<td>Exceptions</td>
<td>11/20</td>
<td>Polymorphism</td>
</tr>
<tr>
<td>9/26</td>
<td>Lab 6: Exception handling</td>
<td>11/21</td>
<td>Lab 12: Final project dissection</td>
</tr>
<tr>
<td>9/27</td>
<td>Files</td>
<td>11/22</td>
<td>Final project discussion</td>
</tr>
<tr>
<td>9/28</td>
<td>PS5 due</td>
<td>11/23</td>
<td><strong>Thanksgiving</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/2</td>
<td>Classes</td>
<td>11/27</td>
<td>Sorting</td>
</tr>
<tr>
<td>10/3</td>
<td>Lab 7: Testing classes with JUnit</td>
<td>11/28</td>
<td>Lab 14: Work on final projects</td>
</tr>
<tr>
<td>10/4</td>
<td>Classes</td>
<td>11/29</td>
<td>Linked Lists</td>
</tr>
<tr>
<td>10/5</td>
<td>PS6 due</td>
<td>11/30</td>
<td>Final project (part 1) due</td>
</tr>
<tr>
<td>10/9</td>
<td><strong>Fall Break</strong></td>
<td>12/4</td>
<td>Algorithm analysis</td>
</tr>
<tr>
<td>10/10</td>
<td><strong>Fall Break</strong></td>
<td>12/5</td>
<td>Lab 15: Sample final discussion</td>
</tr>
<tr>
<td>10/11</td>
<td><strong>Fall Break</strong></td>
<td>12/6</td>
<td>Final exam review</td>
</tr>
<tr>
<td>10/12</td>
<td><strong>Fall Break</strong></td>
<td>12/7</td>
<td>Final project (part 2) due</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12/12</td>
<td><strong>Final Exam (3:30–5:30)</strong></td>
</tr>
</tbody>
</table>
Appeals Procedures

See the Code of Student Rights and Responsibilities, located in the Class Schedule or on the UofU Web site for more details.

Appeals of Grades and other Academic Actions

If a student believes that an academic action is arbitrary or capricious he/she should discuss the action with the involved faculty member and attempt to resolve. If unable to resolve, the student may appeal the action in accordance with the following procedure:

1. Appeal to Department Chair (in writing) within 40 business days; chair must notify student of a decision within 15 days. If faculty member or student disagrees with decision, then,


Withdrawal Procedures

See the Class Schedule or web for more details. ** Please note the difference between the terms “drop” and “withdraw”. Drop implies that the student will not be held financially responsible and a “W” will not be listed on the transcript. Withdraw means that a “W” will appear on the student's transcript and tuition will be charged. **

Drop Period – No Penalty

Students may DROP any class without penalty or permission during the FIRST TEN academic days of the term (Friday, September 1st).

Withdrawal from Full Term Length Classes

Students may WITHDRAW from classes without professor’s permission until Friday, October 20, 2017. Beginning September 2nd until October 20th, a “W” will appear on the transcript AND tuition will be charged. Refer to Class Schedule, Tuition and Fees for tuition information.

Withdrawal from Session I & Session II

See the web page for details: http://registrar.utah.edu/academic-calendars/fall2017.php

Withdrawals after October 20th will only be granted due to compelling, nonacademic emergencies. A petition and supporting documentation must be submitted to the Dean’s Office, 1602 Warnock Engineering Building. Petitions must be received before the last day of classes (December 7, 2017).

Americans with Disabilities Act (ADA)

The University of Utah seeks to provide equal access to its programs, services, and activities for people with disabilities. If you need accommodations in a class, reasonable prior notice needs to be given to the instructor and to the Center for Disability Services, 162 Oplin Union, 581-5020 (V/TDD) to make arrangements for accommodations. All written information in a course can be made available in alternative format with prior notification to the Center for Disability Services.

Repeating Courses

When a College of Engineering class is taken more than once, only the grade for the second attempt is counted. Grades of W, I, or V on the student's record count as having taken the class. Departments enforce these guidelines for other courses as well (e.g., math, physics, biology, chemistry). Attempts of courses taken at transfer institutions count as one attempt. This means a student may take the course only one time at the University of Utah. Courses taken at the University of Utah may not be taken a second time at another institution. If a second attempt is needed, it must be at the University of Utah. Please work with your department advisor to determine the value of repeating courses. Students should note that anyone who takes a required class twice and does not have a satisfactory grade the second time may not be able to graduate. It is the responsibility of the student to work with the department of their major to determine how this policy applies in extenuating circumstances.

Adding Classes

Please read carefully: All classes must be added within 10 academic days of the beginning of the semester (deadline: Friday, September 1st). Late adds will be allowed September 2-September 11, requiring only the instructor's signature. Any request to add a class after September 11th will require signatures from the instructor, department, and Dean, and need to be accompanied by a petition letter to the Dean's office.

A $50 FEE WILL BE ASSESSED BY THE REGISTRAR'S OFFICE FOR ADDING CLASSES AFTER September 11th. ***